REMARKS

It is respectfully submitted that the Examiner has confused "contiguous concatenation of virtual containers" with the concept of "virtual concatenation" as meant by the present invention. In addition, he has misunderstood what the cited reference ITU G707 actually discloses in the paragraphs that he has referred to. In particular paragraph 8.1.7.2, Virtual concatenation of AU-4s, simply states that studies on the concept of virtual concatenation of VC-4s were underway in 1996 but does not offer any suggestion or hint as to what was meant by "virtual concatenation" or how to implement such a concept for VC-4s even if one knew what it meant.

In fact, if one were to adopt the same approach for VC-4s as is mentioned in paragraph 8.3.6.2 of ITU T/G707 for the virtual concatenation of TU-2s in the higher order VC-4, this would lead one away from the present invention to an impractical solution. The method described in paragraph 8.3.6.2 is completely different to what the present invention proposes, as will be explained in detail below.

The present inventor was working on the concept of virtual concatenation at the time with no certainty that such investigations would lead to anything compatible with existing and future SDH networks. There is certainly nothing in any of the citations applied by the Examiner to point a skilled person in the direction of implementing such a concept in the specific manner as proposed by the present invention.

The other relevant parts of the ITUG707 Standard cited by the Examiner are *wholly* concerned with the implementation of *contiguous concatenation of virtual containers* using an SDH Frame of the type shown in Figs. 1-3 of the present application.

Applicant believes that it may be worthwhile explaining for the benefit of the Examiner, some of the basic points about SDH frame structures, virtual containers, contiguous concatenation, and virtual concatenation as proposed by the present invention.

BACKGROUND INFORMATION

It is well known to a person skilled in the art of designing SDH networks, that digital data is transported across an SDH network by loading the data in a plurality of repeating synchronous transport modules (STM) that are also called SDH frames. The time interval of each SDH frame has its origin in the need to retain all the information of an analogue signal (voice signals) when digitizing it. A well-known theorem (Shannon theorem) states that in order to digitize an analogue signal it is necessary to sample the analogue signal at regular intervals and at a frequency higher than twice the signal bandwidth. Therefore if one takes a voice signal whose bandwidth is approximately 50 - 4KHz, it must be sampled at least 8KHz times a second. This gives a repeat time interval of 1/8 times 8 KHz = 125 μs.

This time interval was taken as the basic time interval of an SDH frame when SDH Standards were first set and continues to be the standard that is now incorporated in the ITU T/G707 Standard relied on by the Examiner. The basic data transport rate is taken to be 155.52 Mbit/s, and the data (payload) is accommodated in what is referred to as a container (C). All of this is set out in the G707 standard cited by the Examiner. In accordance with the G707 standard each SDH frame is conventionally represented by 9 rows made up of 270 bytes giving a total of 2430 bytes, and in its basic form this gives a data rate that is 2430 times 8000 / s = 155.52 Mbits/s (sometimes expressed as 155.52 mbps); this is shown in Fig. 1 of the present application.

The first nine columns contain the section overhead (SOH) for transport-support features such as framing, management-operations channels, and error monitoring, with the first segment containing the frame word for demultiplexer alignment. The remaining columns can be assigned in many ways to carry lower bit-rate signals, such as 2 Mbits/s; each signal has 1 byte path overhead (POH) as shown in the drawings of the present application.

The signals to be transported by an SDH network come from connections that can be synchronous, plesiochronous, or asynchronous, and are transported as time-multiplexed packets. To facilitate their transport, one accumulates the packets in "virtual containers" (VC) each of which comprises a container and a specific pointer in the SOH that points to that container. Each VC can extend over a repeating frame, and is the payload entity that travels across the network, being created and dismantled at or near the service termination points (the input port of an ingress node and the output port of an egress node respectively). PDH traffic signals are mapped into containers of appropriate size for the bandwidth required, using single-bit justification to align the clock rates where necessary. Path Overheads (POH) are then added for management purposes, creating a VC, and these overheads are removed later where the VC is dismantled and the original signal is reconstituted at an output port of an egress node.

The first level of division of the frame is the administrative unit (AU), which is the unit of provision for bandwidth in the main network. In capacity the AU can be used to carry a high bit-rate signal, such as 45 Mbits/s or 140 Mbits/s (for the two sizes of AU, AU-3 and AU-4, respectively). VC-4 together with its pointer is referred to as level 4 administrative unit, or AU-4 for short. There are various virtual containers for each type of signals to transmit. For example, for 2 Mb/s and 155 Mb/s, the virtual containers (VC) are the VC-2 and VC-4 respectively.

An AU can be further divided to carry lower-rate signals, each within a tributary unit (TU), of which there are several sizes. For example, a TU-12 carries a single 2 Mbits/s signal, and a TU-2 carries a 6 Mbits/s signal. A specific quantity of one or more TU-ns and AU-ns can be notionally combined into a tributary unit group (TUG) and AUG respectively) for planning and routing purposes. No overheads are attached to create this TUG or AUG, so its existence relies on network management tracking its path.

Telecommunications networks have always been described using the usual model of layers in order to facilitate their design, implementation, and management. On the one hand, the protocols and information exchange units are described within each layer, while on the other hand, the interfaces between the different layers are defined. So it is with SDH architecture.

Each layer communicates with its counterpart at the other end of the transmission by making use of specific overheads that allow one to manage the information carried, detect errors, determine the type of payload and follow up the path established. In the SDH hierarchy, four layers are defined as:

- Two lower layers relating to the network transmission medium. These are the Regeneration Section (RS) and the Multiplex Section (MS). The associated overheads are called RSOH and MSOH respectively.
- Two upper layers relate to the paths of the tributary transported. These are the higher Order Path (35, 45, 140 Mbits/s), and the lower order path (LOP) (1.5, 2, 6, 8 or 34 Mbits/s); the associated overheads are called HO-POH (Higher Order Path Overhead) and LO-POH (Lower Order Path Overhead) respectively.

In this way, SDH segregates its functions into different transport layers, each providing a series of services to the next layer up in the model. In other words, SDH adopts the classic client-server model of division by layers and functions associated to each layer defined.

The physical framework in which the layers are implemented is the collection of telecommunications equipment or network elements (also called nodes). Each element reaches a certain height in the layers and they exchange perfectly standardized units of information with each other.

At each level, subdivisions of capacity can float individually between the payload areas of adjacent frames. This allows for clock differences and wandering as payloads traverse the network and are interchanged and multiplexed with others. In this way, the inevitable imperfections of network synchronization can be accommodated. A pointer that is embedded in the overheads is used to find the floating part of the AU or TU. The AU pointer locates a higher-order VC, and the TU pointer locates a lower-order VC. For example, an AU-3 contains a VC-3 plus a pointer, and a TU-2 contains a VC-2 plus a pointer.

Plesiochronous digital hierarchy (PDH) traffic signals to be mapped into SDH are, by definition, continuous. Each PDH signal is mapped into its own VC, and several VCs of the same nominal size are then multiplexed by byte interleaving into the SDH payload. This arrangement minimizes the delay experienced by each VC. Although, in theory, an ATM traffic signal is made up of discontinuous cells, each 53 bytes long, ATM idle cells are inserted in the gaps between used cells that are inserted by ATM equipment when it is connected to a PDH or SDH interface. This effectively forms a continuous signal. This is then mapped into its own VC-n, just as for a PDH signal, and again multiplexed with other signals by byte interleaving.

The basic STM frame is referred to as STM-1 and has a maximum bit rate of 155 Mbits/s. To form an STM-N synchronous transport module (where N represents a different level N) with higher bit rates than 155Mbits/s from an STM-1, AU-4 byte-interleaved multiplexing is used and a new overhead is added; (this is because the multiplexers take apart the section overheads received and add new overheads to the multiplexed signals they transmit).

Like the STM-1 frame, the STM-N frame is represented as a rectangular structure of 270 x N columns and 9 rows, which gives a total of 270 x N x 9 = 2430 x N bytes. Nonetheless, the frame period remains the same as that of the STM-1 frame, namely, 125μs. The new section overhead (SOH) has a single row x 9 columns taken up by the AU-n pointer (formed by interleaving the pointer bytes of the four VCs) and 8 rows x 9 columns.

Suppose one wants to transmit a broadband signal of hundreds of Mpbs inside an STM-1 frame. The SDH stem does not define optimized containers for transporting tributary signals whose bit rate is higher than 140 Mbits/s. This means that a signal with a transmission rate higher than 140 Mbits/s cannot be packed in any predefined container. For a signal of hundreds of Mbit/s, it is best to associate the structures with the largest capacity available, that is, the VC-4, capable of accommodating signals of up to 140 Mbit/s. How is this done?

<u>CONTIGUOUS CONCATENATION OF VIRTUAL CONTAINERS *(PRIOR ART)*</u>

The "contiguous concatenation" mechanism provides a solution to this problem. It consists of combining contiguously a number of virtual containers needed to form a bigger virtual container (VC) whose capacity is greater than, or equal to, the capacity required. This larger virtual

container is often referred to as a concatenated virtual container or (VCc) (the "c" indicating that it is a concatenated structure).

The contiguous concatenation process is used for forming a virtual container based on several concatenated level 4 virtual containers VC-4c (the letter "c" signifies that the data is concatenated) that takes up the whole STM-4 frame. Note that the structure is analogous to that of an STM-1 frame in relation to its VC-4. The STM-4 frame transports a single virtual container (VC-4), instead of transporting four byte-interleaved VC-4s from other STM-1 frames.

Each VC-4-4c, (the last 4 signifies the fact that four VC-4c are concatenated), has one column reserved for the path overhead (POH) and 3 columns are used for fixed stuffing, leaving 1040 columns for the C-4-4 container. It will be appreciated that such a container has a capacity of 599.040 Mbit/s, enough to transport standard broadband signals of approximately 600Mbits/s).

G707 standardizes the VC-4-4c container for transporting ATM and other broadband signals. and usually considers pointers too, so we talk about AU-4-Xc transported over STM-N. The value of X is only limited by the maximum payload capacity of the STM-N used. For example, where X=4 and N=4; other possible values for X=4 are N=16 and N=64.

Interleaving the bytes of the pointers of the concatenated AU-4 creates a new pointer that points to the AU-4-4c pointer. Before interleaving, the first AU-4 takes the typical values of any conventional AU-4 pointer, while the remaining AU-4 pointers take fixed values in their first bits, to indicate concatenation. VC-4-4c can be transported in STM-16 and STM-64 frames too.

This structure as discussed above is as shown in Figure 8-8 of G707 and Fig. 1 of the present application and is what is referred to as "contiguous concatenation of virtual containers".

All of the references to contiguous concatenation in ITU G 707 define the concatenation of VC-2 and VC-4 virtual containers to create VC-2c and VC-4c respectively.

Contiguous concatenation of virtual containers prior to the present invention uses a concatenation indicator that is present in the pointer associated with each concatenated frame to indicate that the particular VCs with which the pointers are associated, are concatenated. *The resulting VC-4-4c signal only has one path overhead (POH) of 9 bytes and does not have pointers to the individual containers within the VC-4-4vc.* Put another way, prior known concatenation techniques actually strip the path overheads from some of the concatenated containers. As a direct consequence of this some prior known SDH networks at the time the present invention was made would be, and still are, incapable of handling concatenated data structures.

VIRTUAL CONCATENATION (PRESENT INVENTION)

The present invention acknowledges this, and proposes a practical way in which such SDH systems can be modified either by providing them with a new TU card, or by reconfiguring their existing TU cards to operate in a manner never contemplated prior to the present invention, but in accordance with the present invention.

Referring now to "*virtual concatenation*", the G707 states that studies were under way in March 1996, but does not attempt to explain what is meant by "virtual concatenation" compared with the previously known contiguous concatenation, nor does it explain how to implement whatever was meant by "virtual concatenation" at that time. Therefore a skilled person was not directed to do anything different than what had been done before.

Similarly, none of the secondary references relied on by the Examiner explain what is meant by "virtual concatenation" or how to implement virtual concatenation in the way proposed by the present inventor. Applicant will deal with the secondary references later, and explain why a skilled person would find no reason to combine the references in the manner proposed by the Examiner to arrive at the present invention.

From the above description, it will be appreciated that what has been described above *is NOT what the present application calls "virtually concatenated structure"* and that is why the applicant has consistently argued that the Examiner is totally wrong in the Official letter dated June10th 2004, for example in his comments against claim 155, and elsewhere, where he says that Fig. 8-8 of G707 shows "virtual concatenation" as meant by the present invention.

To understand "virtual concatenation" as meant by the present invention the specification teaches that the invention does not simply consist of concatenation of data in the way that has been done before; it involves, as an essential element of the virtual concatenation process, the way of processing the POH of the concatenated VCs *so that the integrity of the path overhead information is maintained.*

That is to say, that prior to the present invention, the path overhead of all but the first VC of concatenated virtual containers (VCc) were stripped off the individual containers that made up each VCc. Consequently some systems could not handle concatenated structures because they relied on pointers that point to every virtual container to be transported. In the present invention, the path overhead of all the VC in the concatenated VCs are effectively recreated and maintained throughout the whole path of the signal thus enabling all SDH systems to handle concatenated data. This is explained in more detail below.

The SDH network of the present invention receives signals for transport across the network at an input port of each ingress node of the network, and converts them into the new virtually concatenated structure in accordance with the present invention. The path overhead is processed in a novel way to create a new "virtually concatenated structure" that has all the path overhead information for not only the concatenated virtual containers, but all of the virtual containers that make up each concatenated virtual container. This enables the signals to be transported across the network like any other concatenated structure while retaining the integrity of path overhead information of the virtual containers in each VCvc.

"Virtual concatenation" takes a number of previously contiguously concatenated virtual containers, for example four VC-4-4cs, that are received at an input port of an ingress node of the network, and converts this first concatenated signal into a second concatenated signal (what the present invention calls a "virtually concatenated structure") that comprises what the present application calls a VC-4-4vc (the first 4 indicating a container in a STM-4 frame, the final 4 signifies that four VCs are concatenated, and the final letters "vc" indicating that the structure is what the present invention calls "virtually concatenated structure"). This is explained in the example on page 6, line 6 of the present application.

During this conversion step, the path overhead is processed to create a pointer that includes a concatenation indicator (to identify that the VCvc has concatenated data) and assigns unique values that points to and identifies each VC within the concatenated VCvc by using the J1 byte of the SOH. The concatenated VCvc is then transported though the network and is eventually converted, at an outlet port of an egress node of the network, back to the same form of concatenated signal as was received at an input of an ingress node.

One should bear in mind that the present invention is not restricted to level four VC-4Cs as described in the example on page 6, and applies to the other TUs and VCs of the SDH architecture as explained at page 8, line 14 to page 9, line 3.

As explained on page 2, lines 9-18, some prior existing SDH networks could not handle concatenated data. The present invention enables these old systems to be modified in accordance with the present invention to transport a concatenated signal over the network simply by utilizing a new TU card at each node of the network or by reconfiguring their existing old tributary cards to work in the same way as a new card.

The new SDH system not only retains the path overhead of all the transported VCs but is backwards and forwards compliant and conforms to the ITUG707 Standard.

Structurally, each ingress/egress node of the SDH network is either provided with a new tributary card, or has its existing TU card re-configured to operate according to the present invention, (as explained at page 5, lines 11-19) so that, in addition to receiving (and outputting) four independent VC-4 signals, it can also handle and process concatenated incoming signals, by creating and maintaining the POH information of the containers within a concatenated virtual container. Put another way, such modified SDH systems would have the additional capability of accepting at its input, and delivering at its output, an STM -4 signal containing four contiguously concatenated VC-4 signals that it could not handle before and convert the signal to one that it can handle while maintaining the path overheads.

The new tributary board has means to recognize the concatenated virtual containers VCc of the incoming first concatenated signal from the "Concatenation Indication" information in

the pointer, and has means for converting the incoming VCc into a "virtually concatenated structure".

For the higher order containers, the path trace byte J1 for each of the VC-ns in each VC-n-Xvc is given a unique code indicating the order within the VC-n-Xvc, the H4 byte is used for frame sequence information (FSI) and the B3 byte is processed to maintain path integrity. In this way, the POH of each virtual container VC-n that makes up each VC-n-Xvc is maintained, and each virtual container VC-n is treated almost individually and subjected to the usual multiplexing and demultiplexing processes.

Secondary features of the present invention provide for aligning the pointers of the VC-ns in the incoming concatenated signal at the input port of an ingress node of the network, and reconstructing and realigning the pointers at the egress node of the network (as described at page 6, lines 4-7 of the present application).

With the present invention, with modern trends requiring bit rates of 2.5Gbps to say 20Gbps, the new SDH system of the present invention can cope with such high bit rates but still remains compatible with, and conforms to the Standards set in 1988 up to March 1996. Furthermore the invention allows older SDH networks that could not handle concatenated signals to handle such signals without requiring extensive modification to existing hardware.

SECONDARY REFERENCE

Turning now to U.S. Patent No. 5,793,760 to Chopping, it will be appreciated that this reference does not contemplate the concept of "virtual concatenation" as discussed above. Therefore even if combined with ITU G707 as suggested by the Examiner, neither of the cited

references, whether taken singularly or in combination, would point one in the direction of the invention claimed. Nor would they point to the specific way of implementing the concept of virtual concatenation as meant by the present invention. This knowledge would only become known in hindsight after reading the present application. In other words, a skilled person would be required to make a further invention over the combination of the cited references in order to arrive at the present invention, and that is what the present inventor has done!

SUMMARY

Summing up, from the above comments, and by reading the present application it will be appreciated that virtually concatenated structures as meant in the present invention are fundamentally different from contiguously concatenated data structures and from virtually concatenated TU-n structures. Contiguously concatenated data structures and virtually concatenated TU-n structures of the prior art rely on the use of the pointer and concatenation indicators in the pointer to point to the VC-ns and actually removes the entire path overhead information from the second and subsequent VC-ns that make up the concatenated virtual container (VC-n-Xc). With the present invention, all path overhead for all the VC-ns is preserved (see page 2, lines 9-18.) The present invention sets forth a practical alternative to handling contiguously concatenated data and, therefore, enables the virtually concatenated data to be carried on existing SDH networks with only minimum modification or reconfiguration of the TU boards.

Allowance of claims 173-215 is respectfully requested. The excess claims fee for five (5) extra independent claims and twenty three (23) extra total claims, is enclosed.

Wherefore, a favorable action is earnestly solicited.

Respectfully submitted,

KIRSCHSTEIN, OTTINGER, ISRAEL & SCHIFFMILLER, P.C.
Attorneys for Applicant(s)
489 Fifth Avenue
New York, New York 10017-6105
Tel: (212) 697-3750
Fax: (212) 949-1690

Alan Israel
Reg. No. 27,564